IOActive Security Advisory

Title	Multiple Buffer Overflows in Legacy mod_jk2 <= 2.0.3-DEV
Severity	High
Date Discovered	05.01.2007
Date Reported	06.27.2007
Date Disclosed	09.20.2007
Authors	Josh Betts, Jason Larsen, Walter Pearce

Affected Products

- mod_jk2 <= v2.0.3-DEV
- F5 BIG-IP <= 9.2.3.30 (Other versions were not tested)

Synopsis

IOActive has discovered a buffer overflow in the Host Header field in the legacy version of the mod_jk2 Apache module (jakarta-tomcat-connectors), which allows for remote code execution in the context of the Apache process.

Description

Versions of mod_jk2 prior to 2.0.4 are vulnerable to multiple stack overflow vulnerabilities. Specifically, IOActive has discovered multiple locations where these vulnerabilities are exploitable via the Host request header in any given request. These overflows all result in remote code execution under the user of the running Apache process. Although a legacy module which is end of life, certain vendors may use this module in their products rendering them vulnerable to remote exploitation

Technical Details

The mod_jk2 module registers with Apache a request handler that parses the entire content of the request—specifically the Host headers—in order to determine which Tomcat worker to forward the request to. For example, multiple buffer overflow opportunities exist within the following code segments:

```
native2\common\jk_uriMap.c: line ~269
      if (port) {
            if (vhost) {
                  if (strchr(vhost, ':'))
                        strcpy(hostname, vhost);
                  else
                        sprintf(hostname, "%s:%d", vhost, port);
            }
            else
                  sprintf(hostname, "*:%d", port);
      }
      else if (vhost)
            strcpy(hostname, vhost);

native2\common\jk_uriMap.c: line ~842
      char key[1024];

      if (!vhost && !port)
            return uriMap->vhosts->get(env, uriMap->vhosts, "*");
      if (!vhost)
            vhost = "*";
      sprintf(key, "%s:%d", vhost, port);
      return uriMap->vhcache->get(env, uriMap->vhcache, key);
```

In each of these code segments, exploitable stack overflows on the Host request header are visible. Additionally, in every circumstance, the condition occurs when a Hostname longer than 1024 characters is provided within the Host: Header request. Exploitation of these overflows is considered trivial.

Remediation

Upgrade to the latest version of the legacy mod_jk2 (mod_jk2 2.0.4) or migrate to the non-legacy reimplementation of this package—the new jakarta-tomcat-connectors—called mod_jk.